CLAIMS

WE CLAIM:

1. A method for determining the capabilities of a media system, the method comprising:
querying each of one or more functional objects in the media system to determine a functional limit of each of the one or more objects for a predetermined function; and
determining which of the functional limits of the one or more objects maximally limits the capability of the media system for the predetermined function..

2. The method of claim 1 wherein the predetermined function is a maximum playback rate of a multimedia stream.

3. The method of claim 2 further comprising determining a minimum of the maximum reported playback rates.

4. The method of claim 2 further comprising determining a minimum and maximum playback rates in a set of modes including: reverse skip mode, reverse key frame mode, reverse full mode, forward full mode, forward key frame mode, forward skip mode

5. The method of claim 1 wherein the one or more functional objects include a media source object, a transform object, and a media sink object.

6. A method for determining whether a playback rate is supported in a multimedia system, the method comprising:
receiving a query for rate support;
retrieving one or more media source components and stream sink components required for a rate change to the playback rate;
for each retrieved media source component, determining if rate control services are supported;
returning an indication to a user as the feasibility of supporting the rate change;
retrieving one or more transforms required for the rate change;
for each transform, calling a rate control service;

for any transform that is a decoder, assuming any rate can be supported; and
if rate control is not supported, assuming any positive rate can be supported.

7. The method of claim 6 further comprising:
notifying a calling application in response to the query for rate support, the notification assuming that a media engine will decode any samples and reverse the samples to perform backward rate changes if a decoder required for the playback rate is not configured to perform backward rate changes.

8. The method of claim 6 wherein the query is received via an application programming interface (API).

9. The method of claim 8 wherein the API is one or more of IMFRateSupport::IsRateSupported, IMFRateSupport::GetMinForwardRate, IMFRateSupport::GetMaxForwardRate, IMFRateSupport::GetMinReverseRate, IMFRateSupport::GetMaxReverseRate, and IMFRateSupport::SetMinForwardRate, IMFRateSupport::SetMaxForwardRate, IMFRateSupport::SetMinReverseRate, IMFRateSupport::SetMaxReverseRate, IMFRateSupport::SetForwardKeyframeRateMultiplier, IMFRateSupport::SetReverseKeyframeRateMultiplier, MFRateSupport::GetForwardKeyframeRateMultiplier, and IMFRateSupport::GetReverseKeyframeRateMultiplier.

10. A method for scheduling a rate change in a multimedia system, the method comprising:
receiving a query to set a playback rate;
receiving one of an identified time for the playback rate or a request for immediate playback rate change;
calling one or more multimedia components required to perform the playback rate; and

scheduling the playback rate using a presentation clock, the presentation clock determining a time to implement the playback rate according to race conditions among the multimedia components.

11. The method of claim 10 wherein the multimedia components atomically schedule the playback rate and return a scheduled time to the presentation clock.

12. The method of claim 11 wherein the multimedia components call a function IMFRateControl::ScheduleRateChange() to schedule the rate change.

13. The method of claim 10 wherein the presentation clock synchronously calls a routine to set a function on a time source wherein calling the routine changes the rate at which the presentation clock is running.

14. The method of claim 13 wherein the routine to set the function on the time source is IMFClockRateSink::OnClockSetRate().

15. The method of claim 10 wherein the presentation clock calls a routine to query one or more clock state sinks.

16. The method of claim 15 wherein the one or more clock state sinks are media sinks configured to operate as renderers.

17. The method of claim 10 further comprising:
waiting for events from the multimedia components indicating that the playback rate has changed; and
upon receiving the events, performing an event indicated that the playback rate has changed to notify a calling application.

18. The method of claim 10 wherein the playback rate change is performed on one of a media engine and a media source.

19. The method of claim 18 wherein the media source performs the playback rate change in response to a set rate call received by a media processor.

20. The method of claim 18 wherein the media source receives data over a network and takes into account a data source before changing the playback rate.

21. The method of claim 18 wherein the media source receives data over a network and notifies a network source of the playback rate change.

22. The method of claim 21 wherein the media source waits for new rate data after receiving media data over the network.

23. The method of claim 21 wherein the media source computes new rate data after receiving media data over the network.

24. A multimedia system comprising:
a control layer configured to receive one or more media data streams from an application; and
a core layer coupled to the control layer, the control layer including a media engine component configured to query each of one or more core layer components in the multimedia system to determine a functional rate limit of each core layer component for a predetermined function, the media engine configured to determine which of the functional limits of the core layer components maximally limits the multimedia system.

25. The multimedia system of claim 24 wherein the core layer includes:
one or more media sources coupled to the control layer, the media sources configured as inputs to the multimedia system;
one or more stream sources coupled to the control layer, the stream sources providing the media data streams;
one or more transforms coupled to the control layer, the transforms configured to operate on the media data streams;
one or more media sinks coupled to the control layer, the media sinks configured to operate as outputs for the media data streams; and
one or more stream sinks coupled to the control layer, the stream sinks configured to store or render the media data streams.

26. The multimedia system of claim 24 wherein the control layer includes:
the media engine;
a topology loader configured to identify data flow;
a media session configured to interface with core layer components; and
a media processor configured to perform transforms on the media data streams.

27. The multimedia system of claim 24 wherein the media engine interacts with a plurality of components in the core layer and the control layer to provide rate changes and rates, the media engine configured to use floating point values to linearly indicate a speed of playback.

28. The multimedia system of claim 27 wherein a negative rate specifies a backward playback.

29. The multimedia system of claim 24 wherein the core layer further includes a media source, the media source configured to provide a presentation timestamp for media samples on the media stream, the samples configured to preserve the presentation timestamp independent of a rate for media playback.

30. The multimedia system of claim 24 wherein the multimedia system further includes a presentation clock configured to run time according to a current rate, and the core layer further includes one or more media sinks coupled to the presentation clock, the media sinks configured to display data according to the presentation clock and independent of non-presentation clock component timestamps.

31. The multimedia system of claim 24 wherein the media engine is configured to respond to requests for rate direction changes by playing out any remaining content up to a timestamp of a direction change, discarding any data in a pipeline, setting a rate of playback and restarting playback.

32. The multimedia system of claim 31 wherein data repeated after the restarting playback is discarded.

33. The multimedia system of claim 31 wherein the media engine is configured to be independent of tracking multiple playback rates unless the rates are within a same mode.

34. The multimedia system of claim 33 wherein one or more components in the core layer are configured to maintain a list of pending rate changes, each component having active only one rate at a time, each component configured to maintain a playback rate independent of tracking rate changes.

35. The multimedia system of claim 24 wherein the media engine is configured to support backward decoding for coder-decoders that do not support backward decoding, the media engine configured to perform forward decoding, and reverse any decoded samples.

36. The multimedia system of claim 35 wherein the reversed decoded samples are available for reuse.

37. A computer-readable medium having computer-executable instructions for performing rate mode changes to a media data stream in a multimedia system, the computer-executable instructions performing acts comprising:
receiving a rate mode change event indicating that a new rate mode;
dropping all data in the media data stream that is waiting to be decoded in a transform pipeline;
marking a first sample of the media data stream to be in the new rate mode with a sample attribute identifying the new rate mode;
changing the rate on a presentation clock according to the new rate mode attribute;
synchronously notifying one or more multimedia components in the multimedia system that provide a time source to perform the new rate mode; and
asynchronously notifying one or more multimedia components in the multimedia system that do not provide the time source to perform the new rate mode.

38. A computer-readable medium having computer-executable instructions for determining the capabilities of a multimedia system, the computer-executable instructions performing acts comprising:

querying each of one or more functional objects in the media system to determine a functional limit of each of the one or more objects for a predetermined function; and determining which of the functional limits of the one or more objects maximally limits the capability of the media system for the predetermined function..

39. The computer-readable medium of claim 38 wherein the predetermined function is a maximum playback rate of a multimedia stream.

40. The computer-readable medium of claim 39 further comprising determining a minimum of the maximum reported playback rates.

41. The computer-readable medium of claim 39 further comprising determining a minimum and maximum playback rates in a set of modes including: reverse skip mode, reverse key frame mode, reverse full mode, forward full mode, forward key frame mode, forward skip mode

42. The computer-readable medium of claim 38 wherein the one or more functional objects include a media source object, a transform object, and a media sink object.

43. A computer-readable medium having computer-executable instructions for determining whether a playback rate is supported in a multimedia system, the computer-executable instructions performing acts comprising:
receiving a query for rate support;
retrieving one or more media source components and stream sink components required for a rate change to the playback rate;
for each retrieved media source component, determining if rate control services are supported;
returning an indication to a user as the feasibility of supporting the rate change;
retrieving one or more transforms required for the rate change;
for each transform, calling a rate control service;
for any transform that is a decoder, assuming any rate can be supported; and
if rate control is not supported, assuming any positive rate can be supported.

44. The computer-readable medium of claim 43 further comprising:

notifying a calling application in response to the query for rate support, the notification assuming that a media engine will decode any samples and reverse the samples to perform backward rate changes if a decoder required for the playback rate is not configured to perform backward rate changes.

45. The computer-readable medium of claim 43 wherein the query is received via an application programming interface (API) the API being one or more of IMFRateSupport::IsRateSupported, IMFRateSupport::GetMinForwardRate, IMFRateSupport::GetMaxForwardRate, IMFRateSupport::GetMinReverseRate, IMFRateSupport::GetMaxReverseRate, IMFRateSupport::SetMinForwardRate, IMFRateSupport::SetMaxForwardRate, IMFRateSupport::SetMinReverseRate, IMFRateSupport::SetMaxReverseRate, IMFRateSupport::SetForwardKeyframeRateMultiplier, IMFRateSupport::SetReverseKeyframeRateMultiplier, IMFRateSupport::GetForwardKeyframeRateMultiplier, and IMFRateSupport::GetReverseKeyframeRateMultiplier

46. The computer-readable medium of claim 43 wherein the application calls an application programming interface (API) to set a multiplier or an intra-frame distance to control the spacing of keyframe processing.

47. A computer-readable medium having computer-executable instructions for scheduling a playback rate in a multimedia system, the computer-executable instructions performing acts comprising:
receiving a query to set a playback rate;
receiving one of an identified time for the playback rate or a request for immediate playback rate;
calling one or more multimedia components required to perform the playback rate; and
scheduling the playback rate using a presentation clock, the presentation clock determining a time to implement the playback rate according to race conditions among the multimedia components.

48. The computer-readable medium of claim 47 wherein the multimedia components atomically schedule the playback rate and return a scheduled time to the presentation clock.

49. The computer-readable medium of claim 48 wherein the multimedia components call a function IMFRateControl::ScheduleRateChange() to schedule the playback rate.

50. The computer-readable medium of claim 47wherein the presentation clock synchronously calls a routine to set a function on a time source wherein calling the routine changes the playback rate at which the presentation clock is running.

51. The computer-readable medium of claim 50 wherein the routine to set the function on the time source is IMFClockRateSink::OnClockSetRate().

52. The computer-readable medium of claim 47 wherein the presentation clock calls a routine to query one or more clock state sinks.

53. The computer-readable medium of claim 52 wherein the one or more clock state sinks are media sinks configured to operate as renderers.

54. The computer-readable medium of claim 47 further comprising:

waiting for events from the multimedia components indicating that the playback rate has occurred; and
upon receiving the events, performing an event indicated that the playback rate has occurred to notify a calling application.

55. The computer-readable medium of claim 47 wherein the playback rate is performed on one of a media engine and a media source.

56. The computer-readable medium of claim 55 wherein the media source performs the playback rate in response to a set rate call received by a media processor.

57. The computer-readable medium of claim 55 wherein the media source receives data over a network and takes into account a data source before providing the playback rate.

58. The computer-readable medium of claim 47 wherein the playback rate is performed on a media source and scheduled to be processed with a media sample at the determined time.

59. The computer-readable medium of claim 47 wherein the playback rate is performed on a media source and associated with a media sample corresponding to the determined time.

60. The computer-readable medium of claim 47 wherein the playback rate is performed on one of the multimedia components and the multimedia component has a media sample corresponding to the determined time, and the multimedia component advances the playback rate through queued data associated with the multimedia component.

61. The computer-readable medium of claim 47 wherein unsent data is regenerated to correspond to the playback rate as the playback rate is advanced through media data.

62. The computer-readable medium of claim 47 wherein unsent data and regenerated data are used to create new data with the playback rate.

63. The computer-readable medium of claim 47 wherein the playback rate includes a direction reversal, and unsent forward data is re-used to generate reverse output data.

64. The computer-readable medium of claim 47 wherein the playback rate is not queued by one of the multimedia components and the playback rate is communicated to a next one of the multimedia components in a pipeline through function calls or associated with a next output sample.

65. The computer-readable medium of claim 64 wherein the next multimedia component compares the playback rate to queued data associated with the next multimedia component, the next multimedia component either processing the playback rate or forwarding the playback rate.